Remarks/Arguments

The Office Action dated March 25, 2004, has been noted, and its contents carefully studied. Reconsideration of the objection to the drawing, and the rejection of the claims under 35 U.S.C. §103, is courteously requested in light of the prior amendments set forth herein and the discussion presented for the Examiner's kind consideration.

Turning to the objection to the drawings, attached is a proposed drawing correction designating Figure 1 as prior art. A further correction is made by identifying one of the two secondary caches with the reference numeral 110 to be consistent with the specification at page 6 in the paragraph spanning lines 5-17. A further amendment has been made to page 8 at line 1 thereof to be consistent with the drawing as proposed to be corrected. In addition, in the specification reference to number 150 has been deleted since no such number appears in the drawing. Finally, the incorrect spelling of "poach" at line 20 of page 8 has been corrected.

In addition to the proposed drawing correction, also attached are copies of the drawing as corrected for use in the application, in the event the Examiner approves the drawing correction. If the Examiner approves the proposed drawing correction, he is courteously requested to enter the attached corrected drawing into the application as a substitute Figure 1 for the originally filed Figure 1.

Turning now to the claims, it is noted that certain amendments have been made, in particular to claims 1 and 11, to more clearly define the method and system of Applicants' invention in a manner which clearly distinguishes over the combination of the cited references. Claims 7 and 8 have been canceled and replaced by new independent claim 21 which recites the subject matter of claim 8, incorporating the subject matter of all intervening claims from which it depends.

With respect to claim 1, an amendment has been made to clearly indicate that the process involves detecting at least one other processor of the plurality of processors which is not idle. In accordance with a further amendment, if the other processor which is not idle remains not idle when the predetermined period of time has elapsed, a process on a queue is then poached to be run by the processor which is idle. A similar amendment has been made to claim 11 to clarify that the poaching occurs at the time the predetermined amount of time expires. The term "process" has a very definite meaning within the specification of this application, and as set forth in page 3, "poaching" is defined as taking a job from another processor or locales ready list, and this is done upon the expiration of the predetermined time.

New method claim 21 clearly requires the starting of a timer at the at least one other processor when it goes not idle, and if said at least one other processor which is not idle remains not idle for the predetermined amount of time, then poaching a processor therefrom is done by an idle processor. As still further recited in claim 21, it requires that the poaching is conducted in a manner where the idle processor first tries to poach from a non-idle processor electrically closest to it, and if the processor electrically closest is idle, trying to poach from the next processor which is electrically closest until encountering a non-idle processor from which poaching can occur. The time period during which a non-idle processor is allowed to remain not idle is greater the farther electrically away a non-idle processor is located relative to an idle processor.

It is respectfully urged that the invention as now recited in the claims is not obvious under 35 U.S.C. §103 from the cited combination of references, as will become more clearly evident from the following detailed discussion of these references presented herein for the Examiner's kind consideration.

U.S. Patent No. 5,745,778 to Alfieri

U.S. Patent No. 5,745,778 to Alfieri (hereinafter "Alfieri") merely describes the prior art of a multiprocessor system having the problems of reduced throughput as a result of what was then apparently an attempt to improve throughput, to which the present invention is an improvement and resolves the reduced throughput problems which occur in a system such as that as Alfieri. In a system, such as Alfieri, an attempt to improve performance in a NUMA system was made by adding additional processors. When performance was tested, it was unexpectedly uncovered that as processors were added through-put actually declined and the number of transactions that could be processed in a minute declined. Analysis revealed that the problem resulted from processors which were not running a user process, i.e., processors which were technically idle, but which as a result of the idle state were actually spinning constantly searching for tasks to process.

An example of the way tasks are arranged in such a system are described in Alfieri. In Alfieri "threads", which are programming constructs that facilitate efficient control of numerous asynchronous tasks, are assigned priorities in a limited portion of a range. In such a system, data structures used by the idle processor are not allocated from local memory, and as the processors were idle in their own data structures, they would actually read and write memory from a far locale, which imposes significant system overhead. In addition, it has been uncovered that the processors search for work too aggressively and upon determining that there is no work to do in their own queue, start to examine other lists for other processors in other locales. In this instance, the processor immediately poaches a process from another processor, even if that other processor is suddenly going to become idle. This causes moving all of the process data and cache footprint to the idle processor, resulting in reduced through-put.

Thus, Applicants' broad concept of only allowing poaching when a predetermined period of time has lapsed, and then poaching a process on a queue is simply not taught or suggested by Alfieri standing alone, or in combination with the hereafter discussed Kimmel et al. reference. It is also noted that the Examiner has specifically acknowledged in paragraph 7 of the Office Action that Alfieri does not specifically teach the use of a predetermined period of time during which at least one processor remains not idle.

U.S. Patent No. 6,105,053 to Kimmel et al.

U.S. Patent No. 6,105,053 to Kimmel et al. (hereinafter "Kimmel") teaches an attempt to improve the system of Alfieri to obtain a balanced processor load across a multiprocessor system such as that of Alfieri, by performing periodic load balancing by promoting all active thread groups to the highest and most visible level in the system architecture. The NUMA system of Alfieri is abstracted in Kimmel so as to manage cost tradeoffs and implement policies and mechanisms that take into account resource access costs while spreading workloads across system resources. (Column 1, lines 43 – 52).

More specifically, Kimmel implements a complicated process which does not even remotely resemble Applicants' invention which involves the measuring of a predetermined time and then poaching a process from a non-idle processor by an idle processor. Instead, Kimmel teaches a complicated method of creating a plurality of thread groups, each with a plurality of threads to implement system balance (see generally column 6).

While Kimmel recognizes that poaching may occur in the system, it is not certain to occur, and only then during very specific instances and under very specific conditions after a much more convoluted process is implemented by the scheduler which involves a number of prior steps. More specifically, the system employs a hierarchical tree. The level in a hierarchical tree is designated as a scheduling level and the nodes in the scheduling level as scheduling locales for the purpose of enabling thread groups to be assigned a home scheduling locale and a current scheduling locale. A thread group may have a different home scheduling locale, than its current scheduling locale and this would occur when a thread group is moved from one scheduling sub-tree to another in order to keep a processor busy that has no eligible thread groups. It is clear however, that in order to improve processor/memory affinity, it is preferable that a thread group have the same home and current scheduling locales because each scheduling locale represents different physical memory and the processors operate more efficiently if they use the local resources (column 10, lines 15 – 33).

At a medium scheduler level, a balance is provided between processor/memory affinity and system throughput by monitoring thread groups to identify languishing thread groups and candidates for load balancing, by monitoring the loads of the respective scheduling locales to identify any load imbalances (column 10, lines 34 – 39). This has nothing to do with Applicants' claimed invention wherein poaching is done irrespective of balances, and it is instead done on a processor searching intensive basis to identify eligible candidates, and possibly poach in accordance with a very simple and specified criteria relating to the predetermined amount of time during which a non-idle processor remains non-idle.

In accordance with Kimmel, when a languishing thread group is identified, three things may occur. First, the thread group may be boosted in priority to increase the likelihood that the processor whose run queue included thread group will select and execute the thread group. Second, the scheduler may promote a thread group to the run queue of a higher node to increase its visibility to other processors. Third, the scheduler may set a poach/help hint flag associated with the thread group to identify the thread group as a good thread group for a processor to either poach or help (column 10, lines 50-60).

In this regard, reference is made to the section at column 14 referred to by the Examiner. It is respectfully urged that what is taught therein is first that the scheduler sets the thread group poach/help hint flag. Because if the dispatcher looking for work in another sub-tree will take a thread group having its poach/help flag set before it takes a thread group not having its poach/help hint flag set, setting the flag increases the likelihood (does not actually poach) that a dispatcher in a different scheduling sub-tree will poach or help the thread group. In no way does this teach or suggest the claimed poaching after a predetermined amount of time has expired. The timing referred to by the Examiner merely relates to the identification of possible candidates for poaching to increase the likelihood. In no event is there any teaching therein of poaching at the time the predetermined amount of time lapses.

Thus, at the broadest levels as now clarified in amended claims 1 and 11, the invention is simply not taught or suggested by the cited combination of references. Moreover, as set forth in new claim 21, the combination of poaching at the time the time elapses, as combined with selection of the electrically closest processors and wherein the timing is conducted by the processor to be poached and allowed by that processor in a local manner once the tide elapses, is just simply not taught or suggested by the cited combination of references.

Moreover, the Examiner has asserted that certain aspects of the invention are obvious, although there is acknowledgement that the prior art fails to teach or suggest such features. The reason behind such rejections is not understood and it is respectfully urged that such a rejection is not permitted under the law. For instance, as to claim 11, the Examiner acknowledges that Kimmel does not specifically teach each processor having a cache associated therewith, but states that it would have been obvious to one skilled in the art to have included a cache. The Examiner acknowledges an improvement in the system by having cache at each processor which increases access to memory for increasing throughput of the entire system, yet fails to point to a reference which would suggest making such a change.

Thus, for the foregoing reasons, it is respectfully urged that all of the claims clearly define patentable subject matter under 35 U.S.C. §103. Nonetheless, should the Examiner have any comments, questions or suggestions of a nature necessary to expedite the prosecution of the application, he is courteously requested to telephone the undersigned at the number listed below.

Dated: June 23, 2004

Respectfully submitted,

A. José Cortina, Reg. No. 29,733

Daniels Daniels & Verdonik, P.A.
P.O. Drawer 12218
Research Triangle Park, NC 27709
Voice 919.544.5444
Fax 919.544.5920
Email jcortina@d2vlaw.com

Enclosures

F:\CL\1301-005\Prosecution\Amendment and Response.doc

PROPOSED DRAWING CORRECTION

Application No. 09/677,120
Amendment dated June 23, 2004
Reply to Office Action dated March 25, 2004
Express Mail EV406652085US
Marked up Replacement Sheet. Figure 1

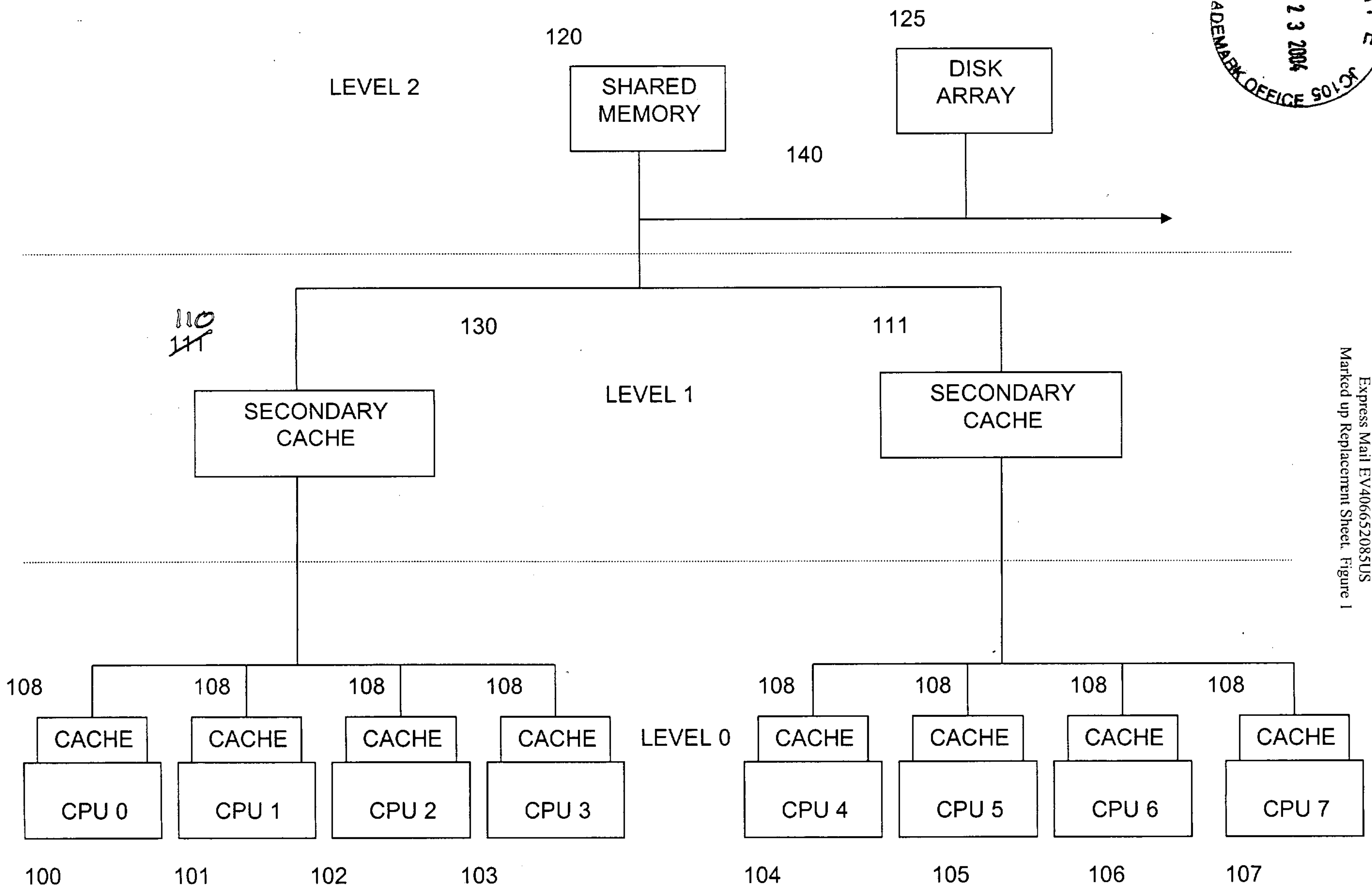

Fig. 1
Prior Art

~~Fig. 1~~